

Claims

- [c1] 1. A rule engine which implements the RETE algorithm with novel extension, which support distributed pattern matching.
- [c2] 2. [Claim Reference] A system according to claim 1 converts an object structure into un-ordered facts within the engine.
- [c3] 3. [Claim Reference] A system according to claim 1 implements the notion of working memory as described in reference article 1.
- [c4] 4. Input nodes as defined in reference article 1 are extended with the capability to route objects to a remote rule engine or use it locally.
- [c5] 5. Join nodes as defined in reference article 1 are extended with the capability to route matched patterns locally or remotely
- [c6] 6. Terminal nodes as defined in reference article 1 are extended with the capability to route matched patterns locally or remotely

- [c7] 7. [Claim Reference] A template according to claim 2 uses a linear sequence of nodes representing the pattern to match for an object.
- [c8] 8. [Claim Reference] A system according to claim 1 implements an abstraction layer for retrieving remote facts.
- [c9] 9. [Claim Reference] A system according to claim 1 uses a call back mechanism between the working memory and the input objects and the object instance uses the call back mechanism to notify the engine when data changes.
- [c10] 10. [Claim Reference] A system according to claim 1 requires object instantiations to implement a base interface for the call back mechanism.
- [c11] 11. [Claim Reference] A system according to claim 1 monitors the resource usage.
- [c12] 12. [Claim Reference] A system according to claim 1 uses rules to manage the distribution of nodes to remote systems.
- [c13] 13. [Claim Reference] A system according to claim 1 distributes pattern matching by serializing the nodes to a remote system.
- [c14] 14. [Claim Reference] A system according to claim 1 dis-

tributes the input, join, terminal and intra-element nodes to a remote system.

- [c15] 15. [Claim Reference] Distributed nodes according to claim 14 maintains a list of remote systems which depend on the results of pattern matching.
- [c16] 16. [Claim Reference] A system according to claim 1 will serialize the values of an object to a remote system if the corresponding pattern matches against a remote object pattern.
- [c17] 17. [Claim Reference] A system according to claim 1 may serialize the object and its values to a remote system if the object contains procedural logic and functional attachments including remote service method calls.
- [c18] 18. [Claim Reference] A system according to claim 1 may serialize the values of an object to a remote system and the receiving system may create a new instance of the object for pattern matching.
- [c19] 19. [Claim Reference] Objects according to claims 16 to 18 are considered temporal by the rule engine if the object's original instance and nodes reside on a remote system.
- [c20] 20. [Claim Reference] A system according to claim 1 de-

defines three types of input channels: standard input, data distribution and node distribution.

[c21] 21. [Claim Reference] A system according to claim 1 defines a set of APIs to handle incoming events and requests for pattern matching.

[c22] 22. [Claim Reference] Input according to claim 21 is defined as standard input channel.

[c23] 23. [Claim Reference] A system according to claim 1 defines a data distribution channel for sending and receiving remote data between rule engines.

[c24] 24. [Claim Reference] A system according to claim 1 defines a pattern distribution channel for distributing RETE nodes.

[c25] 25. [Claim Reference] A system according to claim 1 considers an object as temporal if it was sent through the data distribution channels.

[c26] 26. [Claim Reference] Temporal objects according to claim 19 and 25 are used by the engine to perform pattern matching and these objects are discarded immediately after the pattern matching process is complete.

[c27] 27. [Claim Reference] A system according to claim 1 will route the results of claim 26 back to the originating sys-

tem using the data distribution channel.

- [c28] 28. [Claim Reference] A system according to claim 1 will update the index of the join and terminal nodes as a result of pattern matching according to claim 26.
- [c29] 29. [Claim Reference] A system according to claim 1 processes the RHS of the rule if the original event/request began locally.
- [c30] 30. [Claim Reference] A system according to claim 1 uses messaging system to route new event/request to a cluster of rule engines.
- [c31] 31. [Claim Reference] A messaging system according to claim 30 filters new messages and routes them to the correct engine.
- [c32] 32. [Claim Reference] A system according to claim 1 uses messaging system to route the final result to the recipient.
- [c33] 33. [Claim Reference] A system according to claim 1 contains a component responsible for communicating with the messaging system.
- [c34] 34. [Claim Reference] A messaging component according to claim 33 is responsible for processing inbound events and generating new messages for outbound publication.

- [c35] 35. [Claim Reference] A system according to claim 1 prefers to process new events asynchronously using the messaging system.
- [c36] 36. [Claim Reference] Distributed nodes according to claim 14 contain information about the originating engine, a timestamp of when the nodes were distributed and a priority attribute.
- [c37] 37. [Claim Reference] The priority attribute according to claim 36 may be used by the engine to remove the nodes, if all local object instances have been removed from the working memory.
- [c38] 38. [Claim Reference] Distributed nodes according to claim 14 will not be removed from the local pattern matching network, if data for those patterns is still being used, either in active rules about to fire or in remote procedural attachment calls.
- [c39] 39. [Claim Reference] A system according to claim 1 may forward node distribution messages, if the system does not have sufficient resources.
- [c40] 40. [Claim Reference] A forward message according to claim 19 must retain the location of the originating engine and add the current engine's unique runtime name

to a list of recipients.

- [c41] 41. [Claim Reference] A system according to claim 1 will notify the producer of the node distribution message of success or failure.
- [c42] 42. [Claim Reference] Distributed nodes according to claim 14 may be distributed at deployment time.
- [c43] 43. [Claim Reference] A system according to claim 1 will set an attribute of the input node to indicate that the pattern has been distributed.
- [c44] 44. [Claim Reference] An input node according to claim 43 will maintain a list of the remote systems and the total number of data objects routed remotely.
- [c45] 45. [Claim Reference] A system according to claim 1 may not attempt to distribute nodes that were distributed by another rule engine. Instead, it should notify the originating engine it cannot receive additional data until resources are available.
- [c46] 46. [Claim Reference] A system according to claim 1 may randomly select a remote engine to route data to, if the pattern is distributed to more than 1 engine.